



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/997,056	11/29/2001	Ronnie Ira Chaiken	50037.71US01	4833

27488 7590 01/03/2007  
MERCHANT & GOULD (MICROSOFT)  
P.O. BOX 2903  
MINNEAPOLIS, MN 55402-0903

EXAMINER
----------

FOWLKES, ANDRE R

ART UNIT	PAPER NUMBER
----------	--------------

2192

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/03/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

## Office Action Summary

Application No.

09/997,056

Applicant(s)

CHAIKEN ET AL.

Examiner

Andre R. Fowlkes

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 15 September 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-14, 19 and 20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-14 and 19-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_.

### DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/15/06 has been entered.

2. Claims 1-14 and 19-20 are pending. Claims 1, 4, 11, 19 & 20 have been amended. Claims 15-18 have been canceled.

### ***Claim Rejections - 35 USC § 101***

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 19-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Independent claims 19 and 20 recite a computer readable medium. However, applicant states that "the term computer readable media ... includes both storage and communication media", on p. 8:14-20. Communication media are merely electrical, electromagnetic or optical signals, which are transient in nature and thus are neither

Art Unit: 2192

concrete, tangible nor are they a permanent medium upon which to embody the claimed invention. Since the claims must be given their broadest reasonable interpretation, the invention as claimed reads on both statutory and non-statutory subject matter. Claims that are broad enough to read on both statutory and non-statutory subject matter are considered non-statutory.

The examiner suggests that the applicant replace "computer readable medium" with --computer readable storage medium-- to resolve this issue.

#### ***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-14, 19 & 20 are rejected under 35 U.S.C. 102(b) as being anticipated by Meier, U.S. Patent No. 5,802,371.

As per claim 1, Meier discloses **a computer system for generating metadata for use during stack unwinding** (col. 2:27-47, "When displaying the caller stack of a distributed client/server program that uses Remote Procedure Calls (RPC), the user is provided the capability of viewing the RPC calls in the same manner as normal procedure calls. For example, when a breakpoint is encountered in an RPC server

program while using a debugger for distributed programs, the call stacks for the client and server program are appended together into a single distributed call stack... The distributed call stack may span many programs, threads of execution, and computing machines... The distributed call stack may be used to select a routine from the call stack when setting a context for a tool that displays program state information (i.e. metadata for use during stack unwinding). For example, when the user selects a particular routine from the distributed call stack, the source listing, variables that are currently in scope, and thread executing the routine may be shown"), **comprising:**

- **a plurality of procedures wherein each procedure comprises a sequence of binary instructions** (col. 2:34, "client ... program (i.e. procedures comprising a sequence of binary instructions)"),

- **a runtime for generating unwind data**, (col. 1:41-43, "The tool (i.e. runtime) should also allow the user to display and modify the run-time states (i.e. generate unwind data) of all of those programs in a coherent fashion"), **wherein the unwind data includes a first plurality of blocks of metadata having a first order of blocks, wherein each block of metadata is associated with a corresponding procedure in the plurality of procedures** (col. 2:43-46, "when the user selects a particular routine (i.e. procedure) from the distributed call stack, the source listing, variables that are currently in scope (i.e. a first plurality of metadata associated with the corresponding procedure), and thread executing the routine may be shown (i.e. generated by a runtime)"),

- an unwind rewriter programmed to obtain the unwind data and reorder the first plurality of blocks of metadata to generate a second plurality of blocks of metadata having a second order, wherein the first plurality of blocks are reordered in response to a modification of the sequence of binary instructions within a procedure, (col. 4:59-62, "To verify the proper operation of such a distributed set of client/server programs, the user may wish to debug the distributed set of client/server programs as if they were one single program", in this case, the unwind rewriter (obtains the unwind data and) takes a first plurality of blocks of metadata from each of the distributed client/server programs and merges them to form reordered blocks of metadata), **such that the second plurality of blocks of metadata accurately represents the same runtime semantics as that of the unmodified sequence of binary instructions** (col. 2:43-46, "when the user selects a particular routine from the distributed call stack, the source listing, variables that are currently in scope, and thread executing the routine may be shown (i.e. a second plurality of metadata representing the same runtime semantics as that of the unmodified sequence of binary instructions").

Additionally, the examiner would like to thank the applicant for describing how their invention addresses the problems with the prior art. However the examiner would like to note that the amended claims fail to capture these important and possibly distinguishing features listed on p. 7:23-33 of the arguments.

As per claim 2, the rejection of claim 1 is incorporated and further, Meier discloses that **each block of metadata in the plurality of blocks of metadata includes at least one unwind table and at least one unwind information block** (col. 5:2-3, "The present invention also provides the capability of walking up this distributed call stack", and to walk a call stack, one needs to identify the base/starting basic block of the stack and the current/end basic block stack pointer. The instant application stores this information in the unwind table. Additionally, walking a call stack involves an ordered set of actions over a contiguous region of code (i.e. the information pointed to/contained by the unwind information block)).

As per claim 3, the rejection of claim 2 is incorporated and further, Meier discloses that **the at least one unwind information block includes a region header describing a region of zero length** (col. 5:2-3, "The present invention also provides the capability of walking up this distributed call stack", and to walk a call stack, one needs to identify the base/region header of the stack and the current/end basic block stack pointer.").

As per claims 4 & 5, this is a computer implemented method version of the claimed system discussed above, in claims 1 & 2, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Meier's method of walking-up a call stack for a client/server program (col. 2:27-3:10). And, Meier

discloses "storing (i.e. writing) the call relation (i.e. unwind data) between two parallel processing applications", at col. 1:49-51.

As per claim 6, the rejection of claim 5 is incorporated and further, Meier discloses that **parsing the original unwind data comprises identifying a start basic block and an end basic block of a region associated with the modified binary procedure** (col. 2:50, "walking up a call stack", and to walk a stack, one needs to identify the base/starting basic block of the stack and the current/end basic block stack pointer).

As per claim 7, the rejection of claim 6 is incorporated and further, Meier discloses that **identifying the end basic block of the region further comprises splitting a single basic block into two basic blocks, such that a first basic block ends on a last instruction of the region** (col. 2:50, "walking up a call stack", and when the current stack pointer resides in the middle of a basic block, that location is identified as the end of the call stack, for the purposes of a stack walk).

As per claim 8, the rejection of claim 6 is incorporated and further, Meier discloses that **parsing the original unwind data further comprises identifying an unwind information block associated with a basic block in the original order of the basic blocks that includes a when action description record and establishing a link between the when action description record and the corresponding**



**instruction in the basic block** (col. 2:50, "walking up a call stack", and walking the stack involves determining the information stored by the when action description record).

As per claim 9, the rejection of claim is incorporated and further, Meier discloses that **regenerating new unwind data comprises regenerating new unwind tables and new unwind descriptor records** (col. 5:2-3, "The present invention also provides the capability of walking up this distributed call stack", and to walk a call stack, one needs to identify the base/starting basic block of the stack and the current/end basic block stack pointer. The instant application stores this information in the unwind table. Additionally, walking a call stack involves an ordered set of actions over a contiguous region of code (i.e. the information pointed to/contained by the unwind information block)).

As per claim 10, the rejection of claim is incorporated and further, Meier discloses that **regenerating the new unwind descriptor records further comprises determining if basic blocks identified in a single unwind table associated with the original order of basic blocks are associated with more than one unwind table associated with the current order of basic blocks, and if so, creating a new region header describing a region of zero length** (col. 5:2-3, "The present invention also provides the capability of walking up this distributed call stack", and to walk a call stack, one needs to identify the base/starting basic block of the stack and the current/end

basic block stack pointer. The instant application stores this information in the unwind table. Additionally, walking a call stack involves an ordered set of actions over a contiguous region of code (i.e. the information pointed to/contained by the unwind information block)).

As per claims 11-14, these are a computer implemented method version of the claimed system discussed above, in claims 6-10, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Meier's method of walking-up a call stack for a client/server program, at col. 2:27-3:10, col. 10:8-26, "A method for debugging a distributed computer program comprising a client program executing on a first portion of a distributed data processing system and a server program executing on a second portion of the distributed data processing system, said method comprising the steps of:

- (a) determining a first call stack (i.e. unwind data) on the first portion of the distributed data processing system of the client program;

- (b) determining a second call stack (i.e. additional data from the unwind data) on the second portion of the distributed data processing system of the server program by evaluating remote procedure call (RPC) run-time data structures;

- (c) appending the first call stack and second call stack together on the first portion of the distributed data processing system to form a single distributed call stack (i.e. the rewritten unwind data composed of the reordering of the first group of unwind data)" and col. 4:59-62, "To verify the proper operation of such a distributed set of client/server

programs, the user may wish to debug the distributed set of client/server programs as if they were one single program" (in this case, the unwind data accurately represents the runtime semantics of the binary instructions before the binary instructions were perturbed).

As per claims 19-20 these are computer readable medium versions of the claimed system discussed above, in claims 1-3 and 6, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Meier's method of walking-up a call stack for a client/server program (col. 2:27-3:10).

### ***Response to Arguments***

5. Applicants arguments have been considered but they are not persuasive.

*In the remarks, the applicant has argued substantially that:*

1) Meier does not disclose the following new limitation from claims 1, 4, 11, 19 and 20: modifying the procedure to perturb the binary instructions of the procedure and rewriting unwind data such that the rewritten unwind data accurately represents the runtime semantics of the binary instructions before the binary instructions were perturbed, at p. 9:17-11:4 and p. 12:1-4.

*Examiner's response:*

1) The examiner disagrees with applicant's characterization of the applied art. Meier does disclose modifying the procedure to perturb the binary instructions of the

Art Unit: 2192

procedure and rewriting unwind data such that the rewritten unwind data accurately represents the runtime semantics of the binary instructions before the binary instructions were perturbed, at col. 1:41-4:62, as addressed in the above art rejection. Specifically, Meier discloses at col. 4:59-62, "To verify the proper operation of such a distributed set of client/server programs, the user may wish to debug the distributed set of client/server programs as if they were one single program", in this case, the unwind rewriter (obtains the unwind data and) takes a first plurality of blocks of metadata from each of the distributed client/server programs and merges them to form reordered blocks of metadata representing the runtime semantics of the binary instructions before the binary instructions were perturbed.

*In the remarks, the applicant has argued substantially that:*

2) Meier does not teach further processing of a call stack that has already been generated, at p. 11:21-22.

*Examiner's response:*

2) The examiner disagrees with applicant's characterization of the applied art. Meier does disclose the further processing of a call stack that has already been generated at col. 2:33-35, "the call stacks for the client and server program are appended (i.e. further processed) together", as addressed in the above art rejection.

*In the remarks, the applicant has argued substantially that:*

3) Meier does not teach the combination of “a runtime for generating unwind data” and “an unwind rewriter programmed to obtain the unwind data and reorder the first plurality of blocks of meta data to generate a second plurality of blocks of meta data having a second order,” at p. 11:23-25.

*Examiner's response:*

3) The examiner disagrees with applicant's characterization of the applied art. Meier does disclose the combination of “a runtime for generating unwind data” and “an unwind rewriter programmed to obtain the unwind data and reorder the first plurality of blocks of meta data to generate a second plurality of blocks of meta data having a second order, at col. 1:41-43, “The tool (i.e. runtime) should also allow the user to display and modify the run-time states (i.e. generate unwind data) of all of those programs in a coherent fashion”), and col. 2:43-46, “when the user selects a particular routine (i.e. procedure) from the distributed call stack, the source listing, variables that are currently in scope (i.e. a first plurality of metadata associated with the corresponding procedure), and thread executing the routine may be shown (i.e. generated by a runtime)”, and col. 4:59-62, “To verify the proper operation of such a distributed set of client/server programs, the user may wish to debug the distributed set of client/server programs as if they were one single program”, in this case, the unwind rewriter (obtains the unwind data and) takes a first plurality of blocks of metadata from each of the distributed client/server programs and merges them to form reordered blocks of

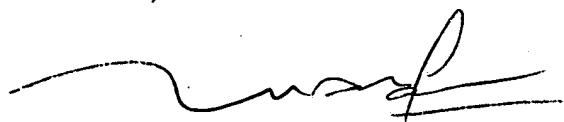
metadata representing the runtime semantics of the binary instructions before the binary instructions were perturbed.

**Conclusion**

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



**TUAN DAM**  
**SUPERVISORY PATENT EXAMINER**

ARF